

**Санкт–Петербургский государственный университет**

***ЧЕРНЫШЕВ Кирилл Владиславович***

**Выпускная квалификационная работа**

***Осесимметричная потеря устойчивости сферических  
оболочек из эластомеров***

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2016 «Прикладная  
математика, фундаментальная информатика и программирование»

Научный руководитель:

доцент, кафедра вычислительных методов  
механики деформируемого тела,  
к.ф. - м.н. Кабриц Сергей Александрович

Рецензент:

ООО "Артех Системы",  
Пронин Андрей Орионович

Санкт-Петербург

2020 г.

# Содержание

<b>Введение</b> . . . . .	3
<b>Постановка задачи</b> . . . . .	3
<b>Обзор литературы</b> . . . . .	4
<b>Глава 1. Математическая постановка задачи</b> . . . . .	6
1.1. Основные зависимости . . . . .	6
1.2. Система уравнений . . . . .	8
<b>Глава 2. Численные методы</b> . . . . .	9
2.1. Метод Ньютона-Канторовича . . . . .	9
2.2. Метод ортогональной прогонки . . . . .	10
2.3. Граничное условие в полюсе . . . . .	12
2.4. Метод продолжение по параметру . . . . .	13
<b>Глава 3. Результаты</b> . . . . .	15
3.1. Жёсткая заделка . . . . .	15
3.2. Шарнирное закрепление . . . . .	16
<b>Выводы</b> . . . . .	18
<b>Список литературы</b> . . . . .	19
<b>Приложение А</b> . . . . .	20
<b>Приложение Б</b> . . . . .	22

## **Введение**

Тонкие сферические оболочки широко используются в кораблестроении, машиностроении, гражданском строительстве и других промышленных приложениях. Помимо этого с помощью теории оболочек можно описывать биологические мембраны, клетки, а также коллоидные частицы. Соответственно, определение их поведения, например, потеря устойчивости при осесимметричной деформации, является важной задачей. Нелинейность уравнений создаёт существенные трудности для их аналитического решения, вследствие чего для исследования применяется комбинация численных методов.

В Главе 1 приводится набор дифференциальных и алгебраических уравнений, соответствующих модели изотропного и несжимаемого материала. Уравнения также относятся и к геометрической постановке - срединной поверхность отвечает дуга окружности, являющегося сечением исходной сферической оболочки.

В Главе 2 описывается алгоритм численного решения поставленной задачи с помощью ряда методов. Также приводится способ обработки граничного условия в полюсе, представляющее особую трудность в случае сферических оболочек.

В Главе 3 приводятся результаты численного эксперимента, в виде диаграмм перемещение-нагрузка в случаях двух граничных условий: 1) жёсткой заделки и 2) шарнирного закрепления.

В приложениях А и Б приводятся листинг части программы, используемой для решения уравнений и имплементирующая описываемые алгоритмы, и формы срединной поверхности для различных состояний рассматриваемых граничных условий.

## **Постановка задачи**

Рассматривается статическая задача потери устойчивости осесимметричной сферической оболочки вращения под действием внешнего давления. Используются уравнения нелинейной теории упругих оболочек, в предположении малости толщины оболочки по сравнению с радиусами кривизны срединной поверхности. Также принимается модифицированная Черныхом

геометрическая гипотеза Кирхгофа[5]: линейное изменение длины материального волокна нормального к срединной поверхности до деформации, остающегося нормальным к ней и после деформации.

Основной целью работы является нахождение состояний устойчивости оболочки в зависимости от различных значений давления, в случае двух граничных условий - жёсткого и шарнирного закрепления по краю. В процессе решения этой задачи также реализуются численные алгоритмы, решающие соответствующую систему уравнений, с помощью языка программирования Python.

## **Обзор литературы**

Поскольку задача состоит в решении механической задачи посредством численных методов, то список литературы можно разделить на 2 блока: литература по механике оболочек и упругих систем и литература по численным методам.

Годуновым в 1961[3] был предложен метод решения краевых задач линейных систем дифференциальных уравнений - метод ортогональной прогонки, позволяющий избегать проблемы "сплющивания" базисных векторов при решении задачи Коши. В [1] Канторовичем был развит метод Ньютона в применении к функциональным пространствам, а также для решения дифференциальных уравнений и краевых задач. Там же доказывается теорема о квадратичной сходимости и даны условия для применимости данного метода. Бахвалов в [2] подробно описывает методы решения дифференциальных уравнений, в том числе методы Рунге-Кутты. Также описывается применение метода Ньютона-Канторовича к краевым задачам, предложен способ решения проблемы восстановления функции посредством исходной нелинейной задачи. В [8] приводится ряд методов для решения нелинейных задач, при параметрической зависимости. Обосновывается метод продолжения по параметру, представляющий существенную основу для численного решения задач и их исследования. В качестве такого параметра для избегания особенностей при построении диаграммы зависимости предлагается использовать длину этой кривой.

В [6] излагается геометрически нелинейная теория упругих оболочек

из эластомеров. В [5] был предложен способ разрешения особенности в полюсе использованием разностной схемы. В статье [7] рассматривается задача осесимметричного выворачивания полусфер из эластомеров в предположении 2 гипотез: Кирхгофа-Черных и Тимошенко-Черных, решение получается с помощью метода множественной стрельбы, в отличие от данной работы, где используется метод квазилинеаризации.

# Глава 1. Математическая постановка задачи

## 1.1 Основные зависимости

Здесь  $\overset{\circ}{\alpha}$  означает характеристику, отнесенную к недеформированной оболочке, просто символ  $\alpha$  относится к деформированному состоянию.

В качестве материальных координат используются длина дуги меридиана  $\overset{\circ}{s}$  и угол  $\overset{\circ}{\theta}$  недеформированной оболочки. Рассмотрим полусферу с центром в начале координат, и вершине в точке  $(0, 0, 1)$ , тогда следующие координаты для точки принадлежащей полусфере:

$$\overset{\circ}{r}' = \sqrt{x^2 + y^2}, \overset{\circ}{x}_3' = z.$$

Принимая обозначение  $' \equiv \frac{d}{d\overset{\circ}{s}}$ , из геометрии следует

$$\overset{\circ}{r}' = \cos \overset{\circ}{\varphi}, \overset{\circ}{x}_3' = -\sin \overset{\circ}{\varphi}.$$

Главные кратности удлинений вдоль окружного и меридианального направлений обозначим как  $\lambda_\theta$  и  $\lambda_s$ , соответственно. Следовательно:

$$\lambda_\theta = \frac{r}{\overset{\circ}{r}}, \lambda_s = \frac{ds}{d\overset{\circ}{s}}. \quad (1)$$

Используя эти зависимости, получаем по аналогии для деформированной конфигурации

$$r' = \lambda_s \cos \varphi, x_3' = -\lambda_s \sin \varphi. \quad (2)$$

В рассматриваемом случае изотропного материала координатные линии являются главными осями деформации и главными осями напряжений, следовательно можно получить следующее выражение для кривизны срединной поверхности

$$\varkappa_s = \varphi' \lambda_s^{-2} \lambda_\theta^{-1} - \overset{\circ}{\varphi}', \varkappa_\theta = \overset{\circ}{r}^{-1} (\lambda_s^{-1} \lambda_\theta^{-2} \sin \varphi - \sin \overset{\circ}{\varphi}). \quad (3)$$

Из общей теории нелинейных оболочек следуют следующие дифференциальные уравнения для усилий и моментов

$$\begin{aligned}(\dot{r}T_s)' - \cos \varphi T_\theta + \dot{r}\varphi' T_{sn} + \dot{r}\lambda_s \lambda_\theta q_s &= 0, \\(\dot{r}T_{sn})' - \sin \varphi T_\theta - \dot{r}\varphi' T_s + \dot{r}\lambda_s \lambda_\theta q_n &= 0, \\(\dot{r}M_s)' - \cos \varphi M_\theta - \dot{r}\lambda_s T_{sn} &= 0.\end{aligned}\tag{4}$$

В данной работе рассматривается равномерное нормальное давление, т.е.  $q_n = \text{const}$ , в то время как  $q_s = 0$ . Также для трёхконстантного несжимаемого материала принимаются верными следующие соотношения для усилий и моментов:

$$\begin{aligned}T_s &\approx \mu \dot{h} n^{-1} \lambda_s^{-1} [1 + \beta + (1 - \beta) \lambda_\theta^n] (\lambda_s^n - \lambda_s^{-n} \lambda_\theta^{-n}), \\T_\theta &\approx \mu \dot{h} n^{-1} \lambda_\theta^{-1} [1 + \beta + (1 - \beta) \lambda_s^n] (\lambda_\theta^n - \lambda_s^{-n} \lambda_\theta^{-n}), \\M_s &= \frac{1}{6} \mu \dot{h}^3 n^{-1} \lambda_s^{-2} \lambda_\theta^{-1} [(1 + \beta) \lambda_s^{-n} \lambda_\theta^{-n} + (1 - \beta) \lambda_s^n \lambda_\theta^n] (\varkappa_s + \frac{\varkappa_\theta}{2}), \\M_\theta &= \frac{1}{6} \mu \dot{h}^3 n^{-1} \lambda_\theta^{-2} \lambda_s^{-1} [(1 + \beta) \lambda_s^{-n} \lambda_\theta^{-n} + (1 - \beta) \lambda_s^n \lambda_\theta^n] (\varkappa_\theta + \frac{\varkappa_s}{2})\end{aligned}\tag{5}$$

, где  $\dot{h}$  - толщина недеформированной оболочки,  $n = 2$  и  $\beta = 1$  в силу свойств рассматриваемого материала.

Итого имеем 6 дифференциальных и 6 алгебраических соотношений, к ним также присоединяются граничные условия:

- в полюсе:

$$\dot{r}(0) - r(0) = 0, \dot{\varphi}(0) - \varphi(0) = 0, T_{sn}(0) = 0.$$

-в первом случае жесткая заделка на крае:

$$\dot{r}(L) - r(L) = 0, \dot{\varphi}(L) - \varphi(L) = 0, x_3(L) - \dot{x}_3(L) = 0.$$

-во втором случае шарнирное закрепление на крае:

$$\dot{r}(L) - r(L) = 0, M_s(L) = 0, x_3(L) - \dot{x}_3(L) = 0.$$

При дальнейшем рассмотрении уравнений, используются перемещения:

$$\rho(s) = r(s) - \mathring{r}(s), \psi(s) = \varphi(s) - \mathring{\varphi}(s), z(s) = x_3(s) - \mathring{x}_3(s).$$

## 1.2 Система уравнений

На основании предыдущих уравнений введем следующие обозначения:  $V = (\mathring{r}T_s, \mathring{r}T_{sn}, \mathring{r}M_s, \psi, \rho, z)$  - вектор компонент выражаемых через дифференциальные уравнения (4),(2),(3),  $W = (\lambda_\theta, \lambda_s, T_\theta, M_\theta, \varkappa_s, \varkappa_\theta)$  - вектор компонент выражаемых через алгебраические уравнения (3),(1),(5). Следовательно уравнения можно записать в следующем виде:

$$\begin{cases} \frac{dV}{ds} = g(V, W, s), \\ f(V, W, s) = 0. \end{cases} \quad (6)$$

Заметим, что если известен вектор  $V$ , то найдя  $\lambda_s$  из первого уравнения в (2), остальные компоненты вектора  $W$  можно найти решением линейных уравнений, последовательно подставляя уже найденные значения. Действительно,  $\lambda_\theta$  находится из линейного уравнения (1), далее подставив в (5) относительно извешного  $T_s$ , и решив данное нелинейное уравнение получаем  $\lambda_s$ , а следовательно и  $T_\theta$ , далее из (3) выражается  $\varkappa_\theta$ , а затем  $\varkappa_s$  из (5), поскольку  $\mathring{r}M_s$  известно, последним получаем  $M_\theta$  подставив найденные компоненты.



## Глава 2. Численные методы

Решение системы (6) осуществляется следующим образом: исходная нелинейные уравнения с помощью метода Ньютона-Канторовича к линейному виду, помимо этого согласно предыдущей главе приходится решать только одно скалярное нелинейное алгебраическое уравнение, остальные пять уравнений линейные. С помощью обработки граничного условия переходим от отрезка  $[0; L]$  к  $[\Delta; L]$ . Получающаяся линейная краевая задача решается с помощью метода ортогональной прогонки, в практической реализации используется метод Рунге-Кутты 4-го порядка для решения задачи Коши. При этом применяется метод продолжения по параметру давления  $q_n$ . Все упомянутые методы описываются ниже.

### 2.1 Метод Ньютона-Канторовича

В [2], [8] описывается метод сведения исходного нелинейного дифференциального уравнения к линейному. Рассмотрим краевую задачу для нелинейного дифференциального уравнение:

$$\begin{aligned}\frac{dy}{dx} &= f(x, y) \\ B(y(0)) &= 0, \\ C(y(L)) &= 0.\end{aligned}\tag{7}$$

Далее строится итеративный процесс, основанный на методе Ньютона, где линеаризации происходит в пространстве функций и основана на матрице Якоби:

$$\begin{aligned}\frac{dy_{n+1}}{dx} &= \frac{dy_n}{dx} + f_y(x, y_n(x))(y_{n+1} - y_n) = f(x, y_n(x)) + f_y(x, y_n(x))(y_{n+1} - y_n) \\ B_y(y_n(0))(y_{n+1}(0) - y_n(0)) + B(y_n(0)) &= 0, \\ C_y(y_n(L))(y_{n+1}(L) - y_n(L)) + C(y_n(L)) &= 0.\end{aligned}\tag{8}$$

Здесь  $f_y(x, y) = [\frac{\partial f_j(x, y)}{\partial y_i}]$ , аналогично для  $B_y$  и  $C_y$ . Данный итерационный процесс происходит в пространстве функций, однако при решении

решения задачи Коши, например методами Рунге-Кутты, необходимо находить значения между точками интегрирования. Эта задача восстановления значений функции решается посредством решения задачи Коши относительно исходной нелинейной задачи для предыдущей итерации, т.е. при решении уравнений (8), вместо вектора  $y_n(x_k)$  в точке интегрирования  $x_k$  может быть использовано  $G(y_n(x_{k-1}, x_{k-1}))$  - решение задачи Коши в точке  $x_k$  относительно уравнений (7) и начального значения  $y_n(x_{k-1})$ .

В нашем случае, исходные уравнения (6) после линеаризации выглядят следующим образом:

$$\begin{cases} \frac{dV_n}{ds} = \frac{\partial g}{\partial V}(V_n - V_0) + \frac{\partial g}{\partial W}(W_n - W_0) + g(V_0, W_0, s), \\ \frac{\partial f}{\partial V}(V_n - V_0) + \frac{\partial f}{\partial W}(W_n - W_0) + f(V_0, W_0, s) = 0 \end{cases}$$

выражая  $(W_n - W_0)$  из второго уравнения, окончательно:

$$\begin{cases} \frac{dV_n}{ds} = \left( \frac{\partial g}{\partial V} - \frac{\partial g}{\partial W} \frac{\partial f}{\partial W}^{-1} \frac{\partial f}{\partial V} \right) (V_n - V_0) - \frac{\partial g}{\partial W} \frac{\partial f}{\partial W}^{-1} f(V_0, W_0, s) + g(V_0, W_0, s), \\ (W_n - W_0) = -\frac{\partial f}{\partial W}^{-1} \left( \frac{\partial f}{\partial V}(V_n - V_0) + f(V_0, W_0, s) \right) \end{cases}$$

В данном случае, все матрицы производных вычисляются относительно предыдущей итерации, обозначенной  $V_0, W_0$ . К полученной системе применимы методы решения линейных дифференциальных уравнений, одним из которых является метод ортогональной прогонки.

## 2.2 Метод ортогональной прогонки

Метод был впервые описан Годуновым в [3], применяется для систем вида

$$\frac{dy}{dx} = A(x)y + f(x), \quad x \in [0; L]$$

Кравые условия имеют следующий вид

$$By(0) = 0; Cy(L) = 0,$$

где  $B$  и  $C$  прямоугольные матрицы, размерностей  $[n - k; n]$  и  $[k; n]$  соответственно.

Исходный отрезок  $[0; L]$  разбивается на  $T$  участков точками  $0 = x_0 < x_1 < \dots < x_T = L$ , также берётся набор линейно независимых векторов  $R^0 = \{y_1, y_2, \dots, y_k\}$ , удовлетворяющих граничному условию в  $x = 0$ , а также вектор  $y_{k+1}^0 = 0$ . На каждом участке интегрирования  $[x_s, x_{s+1}]$  решается следующая система

$$\begin{cases} \frac{dy_j^s}{dx} = A(x)y_j^s, j = \overline{1, k}, \\ \frac{dy_{k+1}^s}{dx} = A(x)y_{k+1}^s + f(x) \end{cases}$$

Обозначим  $Y^s = R^s \cup y_{k+1}^s$ , а также  $K_s(Y^s)$  - решение задачи Коши на отрезке  $[x_s; x_{s+1}]$ . Тогда набор векторов  $Y^{s+1}$  получается из  $Y^s$ , ортогонализацией  $K_s(Y^s)$  методом Грамма-Шмидта. В процессе ортогонализации получаются матрицы перехода  $\Omega^{(s)}$ , связывающие векторы  $K_s(Y^s)$  и новый базис  $Y^{s+1}$ .

В конце отрезка интегрирования  $x = L$  получается набор  $Y^T = R^T \cup y_{k+1}^T$ , относительно  $R^T$  решается задача удовлетворения граничному условию:

$$y(L) = y_{k+1}^T(L) + \sum_{i=1}^k \beta_i^T y_i^T(L)$$

$$Cy(L) = 0$$

Набор получающихся значений  $\beta^T = (\beta_1^T, \dots, \beta_k^T)$  используется для последовательного нахождения значений искомой функции, в процессе обратного прохода от конца отрезка  $L$  к  $0$ :

$$\beta^{s+1} = \Omega^{(s)} \beta^s$$

$$y(x_s) = y_{k+1}^s(x_s) + \sum_{i=1}^k \beta_i^s y_i^s(x_s)$$

Недостаток метода - хранение матриц  $\Omega^{(s)}$  в памяти для всех узлов ортогонализации  $x_s$ . При этом применение данного способа решения линейных краевых задач позволяет избегать проблемы "сплющивания базиса", возникающей, например, в случае, когда действительные части собственных чисел матрицы  $A(x)$  имеют разные знаки.

### 2.3 Граничное условие в полюсе

В случае сферической оболочки в уравнениях (6) в точке  $s = 0$  возникает неопределённость  $\frac{0}{0}$ , что критично для численного решения задачи. Замена исходной граничной задачи и предположение выполнения равенств вблизи полюса позволяет преодолеть данную трудность [5]. Рассмотрим уравнения:

$$r' = \lambda_s \cos \varphi$$

$$r = \lambda_\theta \dot{r}$$

При  $s = 0$  в силу геометрии  $\dot{r}(0) = \sin(0) = 0$ , в силу ограничений в полюсе  $r(0) = \dot{r}(0) = 0$ ,  $\varphi(0) = \dot{\varphi}(0) = 0$ . Тогда, используя правило Лопиталья, получаем следующее тождество для кратностей удлинения:

$$\lambda_\theta(0) = \frac{r(0)}{\dot{r}(0)} = \frac{r'(0)}{\dot{r}'(0)} = \frac{\lambda_s(0) \cos(\varphi(0))}{\cos(0)} = \lambda_s(0)$$

Следовательно равны усилия  $T_s(0) = T_\theta$ , а также используя равенство в нуле усилия  $T_{sn} = 0$ :

$$(\dot{r} M_s)' = \dot{r}' M_s + \dot{r} M_s' = \cos(0) M_s = M_\theta \cos(0) + \dot{r} T_{sn}(0) \lambda_s = M_\theta \cos(0)$$

Значит  $M_s(0) = M_\theta(0)$ , а следовательно равны и кривизны  $\varkappa_s(0) = \varkappa_\theta(0)$ .

В окрестности полюса используем разностную схему для компонент  $\bar{V}(\Delta) = (\dot{r}T_{sn}, \psi, \rho)$  из системы (6):

$$\begin{cases} \bar{V}(0) = \bar{V}(\Delta) - \Delta \bar{g}(\bar{V}(\frac{\Delta}{2}), \bar{W}(\frac{\Delta}{2}), \frac{\Delta}{2}), \\ \bar{f}(\bar{V}(\frac{\Delta}{2}), \bar{W}(\frac{\Delta}{2}), \frac{\Delta}{2}) = 0 \end{cases}$$

Принимая  $\bar{V}(\frac{\Delta}{2}) = \frac{\bar{V}(\Delta) + \bar{V}(0)}{2}$ , из граничных условий следует  $\frac{\bar{V}(\Delta) + \bar{V}(0)}{2} = \frac{\bar{V}(\Delta)}{2}$ , тогда вместо исходных граничных условий и равенств в полюсе, принимаемых верными и в точке  $s = \Delta$ , имеем (индексы опускаются ввиду равенства):

$$\begin{cases} \frac{\psi(\Delta)}{\Delta} = \lambda^3(\varkappa + 1) - 1, \\ \frac{\rho(\Delta)}{\Delta} = \lambda \cos(\frac{\psi(\Delta)}{2} + \frac{\Delta}{2}) - \cos(\frac{\Delta}{2}), \\ \frac{\dot{r}T_{sn}(\Delta)}{\Delta} = T \sin(\frac{\psi(\Delta)}{2} + \frac{\Delta}{2}) - \frac{\dot{r}T(\Delta)}{2} \lambda^3(\varkappa + 1) - \dot{r} \lambda^2 q_n, \\ \bar{f}(\frac{\bar{V}(\Delta)}{2}, \bar{W}(\frac{\Delta}{2}), \frac{\Delta}{2}) = 0 \end{cases}$$

Таким образом исходные граничные условия в полюсе заменяются системой, а отрезком интегрирования в такой постановке является  $[\Delta; L]$ . При этом  $\Delta$  должно выбираться таким образом чтобы погрешность решения разностной схемы не превышала погрешность решения задачи Коши.

## 2.4 Метод продолжение по параметру

Нелинейность и неединственность решения краевой задачи (6) в зависимости от давления  $q_n$  побуждает использовать метод продолжения по параметру. Метод заключается в нахождении решения  $y_1$  некоторого уравнения, зависящего от параметра  $q$ , с учётом известного решения  $y_0$  при значении  $q_0$ :

$$y_1 = f(x, y_1, q_0 + \Delta q), \text{ при } y_0 = f(x, y_0, q_0)$$

,при этом предполагается непрерывность функции  $f$  по  $q$ .

В случае рассматриваемой задачи известно решение при отсутствии давления, изменение параметра осуществляется с помощью постоянного инкремента  $\Delta$ . При достижении некоторых особых точек, увеличение в которых более становится невозможным, продолжить решение можно пытаясь уменьшить инкремент  $\Delta_1 = \frac{\Delta}{2}$ , или же меняя рассматриваемый параметр.

Легитимность метода продолжения по параметру основывается на теореме о неявной функции[8], предполагаемой выполненной в данной задаче.

## Глава 3. Результаты

Состояния равновесия системы определяется двумя параметрами - перемещением полюса оболочки  $z(0)$  и внешним давлением  $P$ [4]. В качестве начального значения-нулевой итерации принимается недеформированное состояние оболочки  $V_0 = (0, 0, 0, 0, 0, 0)$ ,  $W_0 = (1, 1, 0, 0, 0, 0)$ , а параметр  $q_n = 0$ . При увеличении давления может произойти скачок - одному значению  $q_n$  соответствуют более одного значения прогиба  $z(0)$ . Для нахождения совокупности значений отвечающих этой ветви приходится менять параметр давления  $q_n$  на параметр прогиба  $z(0)$ . Однако даже такой способ может быть неэффективен в ряде случаев, тогда придётся искать другой параметр, например момент  $M_s(0)$ .

В обоих рассматриваемых случаях граничных условий радиус оболочки  $R = 1$ , толщина оболочки  $h = \frac{R}{10}$ , длина меридиана  $L = 1.5$ , при дальнейшем изложении давление обезразмеривается  $p = \frac{q_n}{\mu} (\frac{R}{h})^2$ ,  $z(0)$  меняет знак.

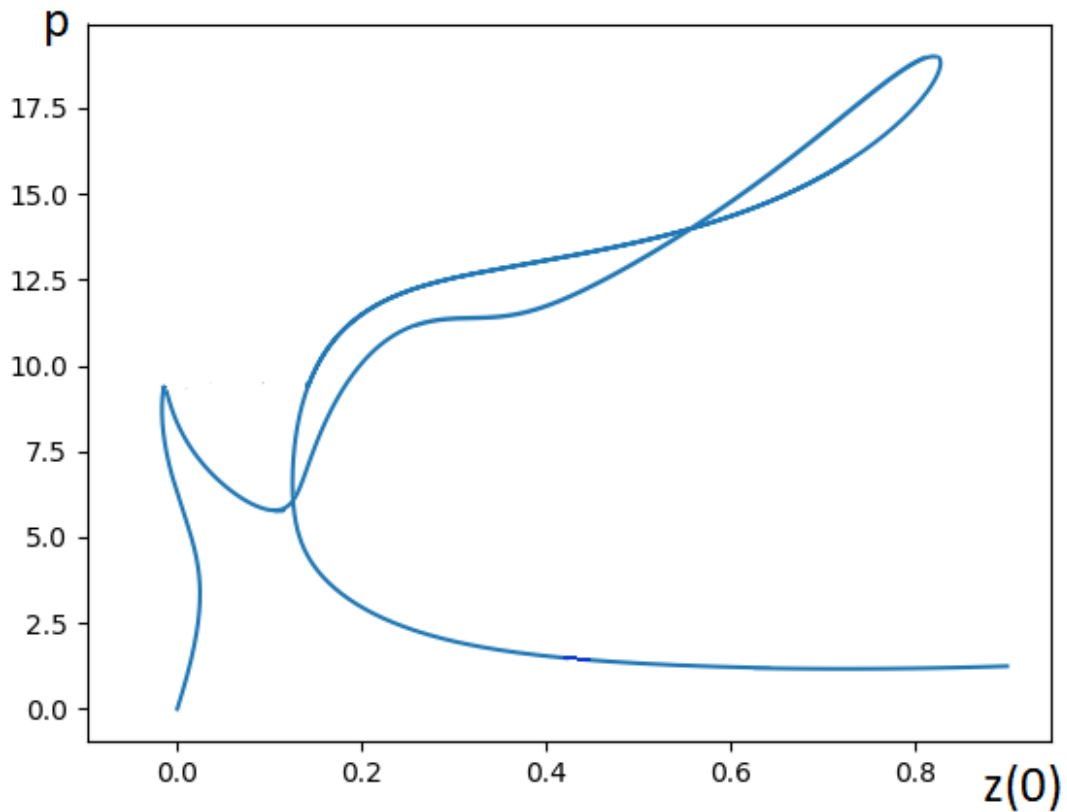
### 3.1 Жёсткая заделка

Даиграмма для данного случая представлена на Рис.1 В точке  $p_1 = 9.2$  находится критическая точка, при увеличении наблюдается перескок из  $z_1(0) = 0.01$  в  $z_2(0) = 0.17$  и при уменьшении давления в этом новом состоянии не наблюдается возвращение назад в точку  $z_0(0) = 0$ , напротив, прогиб полюса оболочки не становится меньше 0.15. Значение давления  $p_3 = 5.78$  также является критической точкой, для её прохождения использовался параметр прогиба  $z(0)$ , а давление  $p$  принимается одной из компонент вектора  $V$ , отвечающая дифференциальному уравнению:

$$\frac{dp}{ds} = 0 ; p(0) = \bar{p}$$

При давлении  $p > 15$  находится "петля", где находятся критические значения и относительно параметра  $p$ , и относительно параметра  $z(0)$ .

В приложении А на фиг.4-13 представлены формы срединной поверхности оболочки в для состояний, обозначенных на диаграмме.



**Рис. 1:** Диаграмма  $z(0) - p$  для жёсткой заделки.

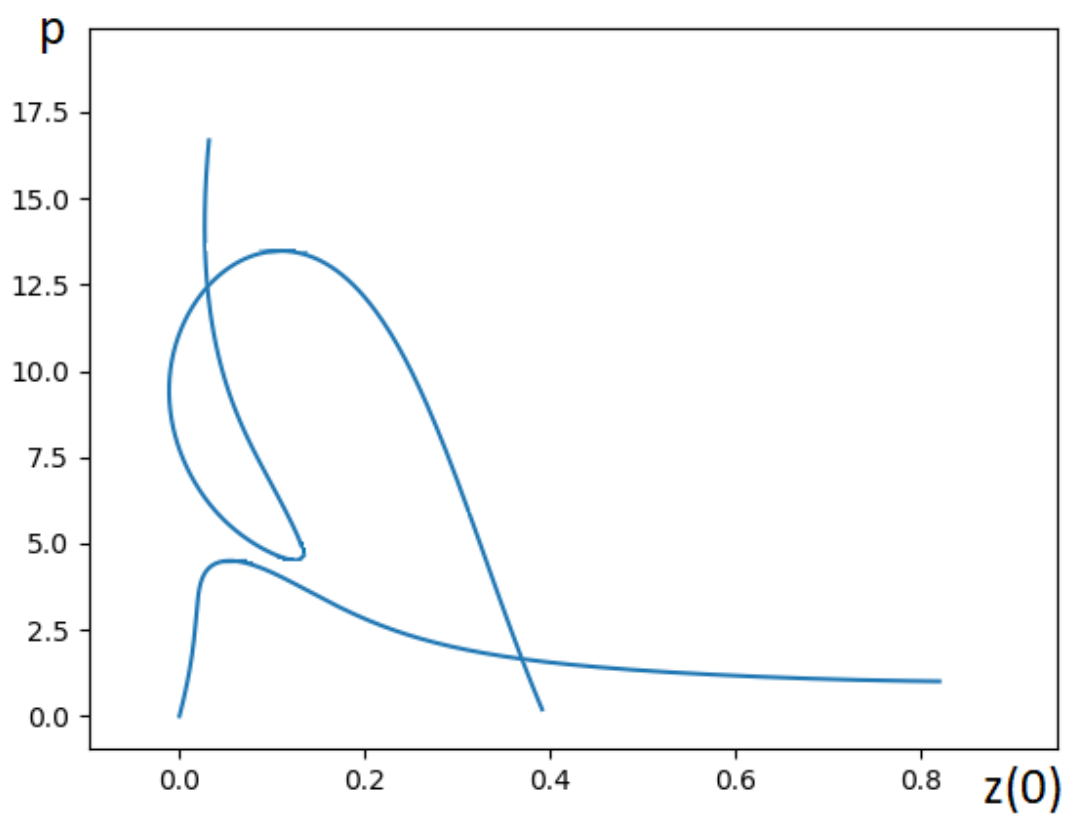
### 3.2 Шарнирное закрепление

В случае шарнирного закрепления можно наблюдать перескок при давлении  $p = 4.6$ , уменьшение давления в таком случае приведёт к опусканию полюса оболочки, а увеличение к попаданию на другую ветвь решения. Как можно заметить следующее критическое значение есть  $p_2 = 13.2$ , уменьшение давления до 0 приводит в случае его достижения к прогибу оболочки с максимальным значением  $z_2^{max}(0) = 0.4$ . Видно, что случай шарнирного закрепления отличается от жёсткой заделки, поскольку в предыдущем случае значению давления  $p = 1.2$  отвечают 2 значения прогиба  $z(0)$ , здесь же ему отвечают 3 значения. Также в отличие от жёсткой заделки максимальное давление, для которого было возможно найти решение  $p_3 = 40.1$ , являющееся критической точкой.

Соответствующие формы срединной поверхности при различных со-



стояниях представлены в приложении А на фиг.15-20.



**Рис. 2:** Диаграмма  $z(0) - p$  для шарнирного закрепления.

## Выводы

Построен и реализован алгоритм решения нелинейных уравнений для осесимметричной деформации сферической оболочки вращения. Существующая имплементация соединяет метод Ньютона-Канторовича, метод ортогональной прогонки и метод продолжения по параметру.

С помощью данных средств были найдены точки потери устойчивости оболочки и диаграмма "прогиб полюса-внешнее давление". Приведены формы срединной поверхности оболочки при некоторых значениях внешнего давления.

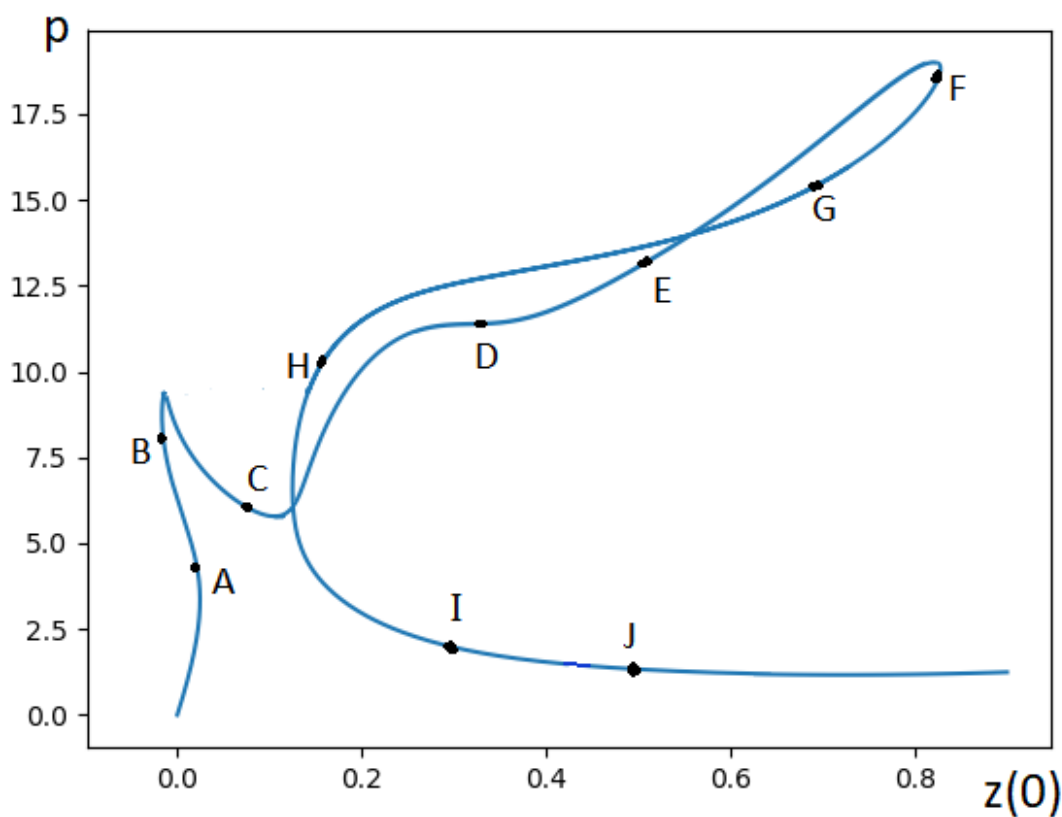
Предложенный алгоритм и полученные результаты могут быть в дальнейшем использованы при рассмотрении оболочек с меньшими толщинами, отличных от рассмотренных граничными условиями или оболочек с эллиптической формой.

## Список литературы

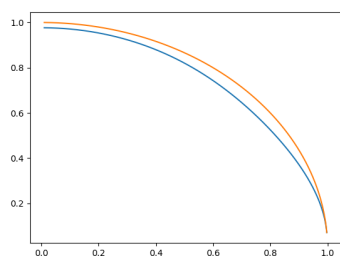
- [1] Акилов Г.П., Канторович Л.В. «Функциональный анализ». М.: Наука, 1984. 752 с.
- [2] Бахвалов Н. С. «Численные методы (анализ, алгебра, обыкновенные дифференциальные уравнения)». М.: Наука, 1975. 631 с.
- [3] Годунов С.К. «О численном решении краевых задач для систем линейных обыкновенных дифференциальных уравнений». Успехи математических наук. 1961. Т. 26, вып. 3 (99). С. 171-174.
- [4] Губанова И.И, Пановко Я.Г.. «Устойчивость и колебания упругих систем». М.: Наука, 1987. 352 с.
- [5] Кабриц С.А. «Некоторые прикладные задачи статики тонких оболочек из эластомеров»: дис. канд. ф.-м. наук: 01.02.04 - ЛГУ, Ленинград, 1984 - 174 с.
- [6] Кабриц С.А., Михайловский Е.И., Товстик П.Е., Черных К.Ф., Шамина В.А. «Общая нелинейная теория упругих оболочек». СПб.: СПбГУ, 2002. 388 с.
- [7] Kabrits S.A., Kolpak E.P. «Quasi-Static Axisymmetric Eversion Hemispherical Domes Made of Elastomers». AIP Conference Proceedings, 2016 - Volume 138, Issue 1 - 160006
- [8] Когай В.В, Фадеев С.И. «Нелинейные краевые задачи для систем обыкновенных дифференциальных уравнений на конечном отрезке». Механико-математический факультет НГУ, 2008. 102 с.

## Приложение А

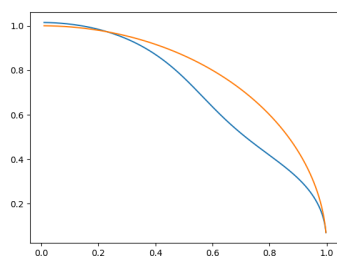
Ниже представлены диаграммы "прогиб полюса - давление" и формы срединной поверхности для некоторых состояний.



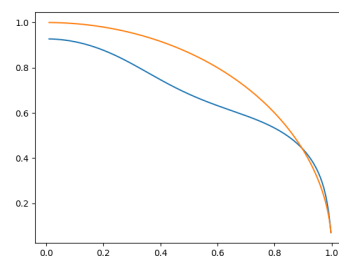
**Рис. 3:** Диаграмма  $z(0) - p$  для жёсткой заделки.



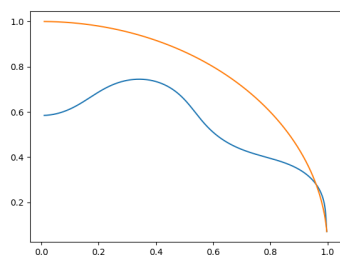
**Рис. 4:** Состояние А



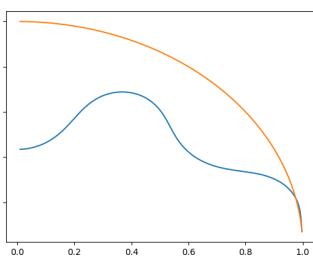
**Рис. 5:** Состояние В



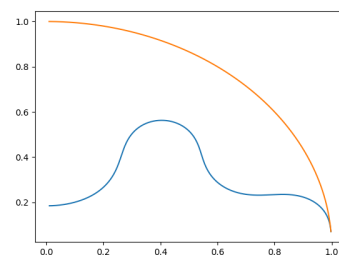
**Рис. 6:** Состояние С



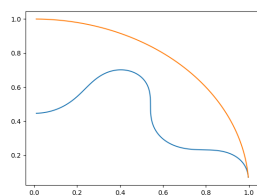
**Рис. 7:** Состояние D



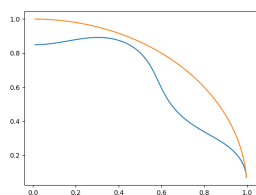
**Рис. 8:** Состояние E



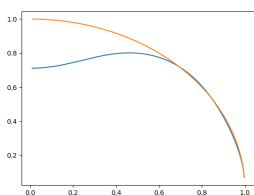
**Рис. 9:** Состояние F



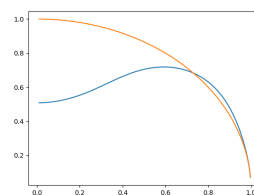
**Рис. 10:** Состояние G



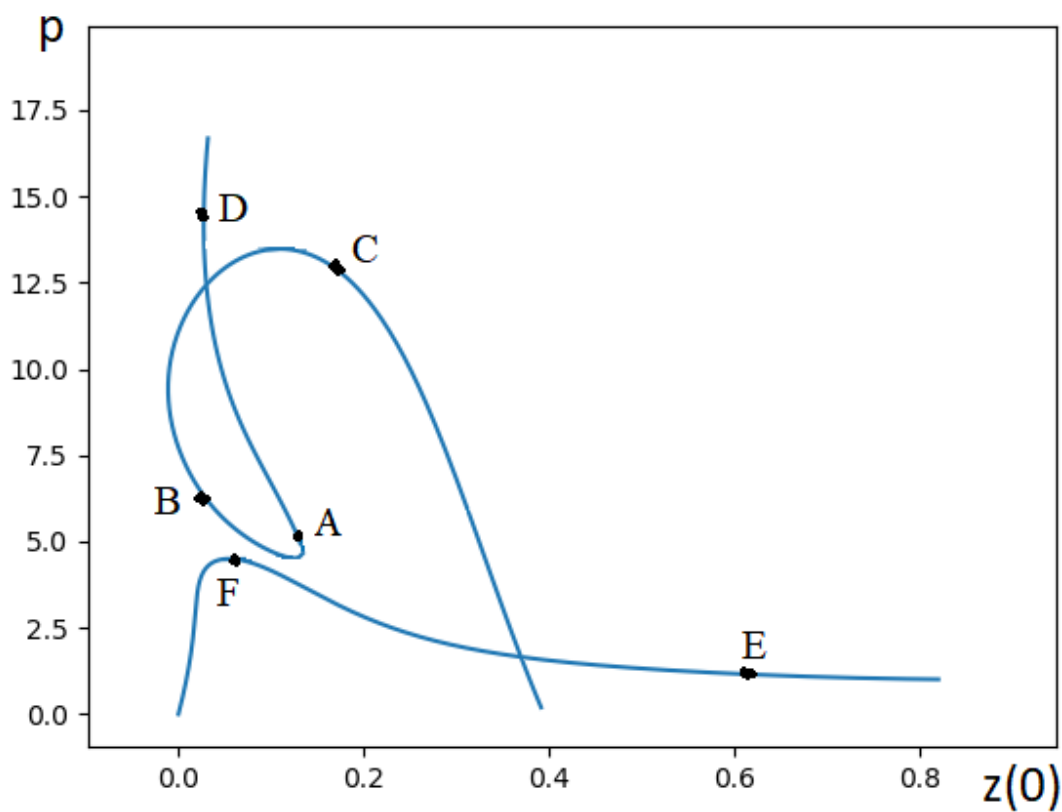
**Рис. 11:** Состояние H



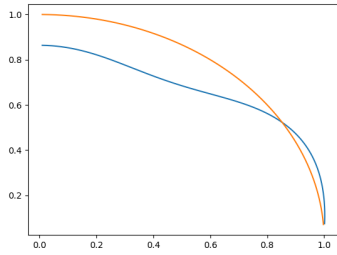
**Рис. 12:** Состояние I



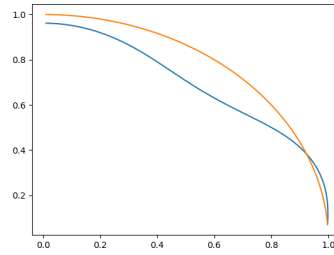
**Рис. 13:** Состояние J



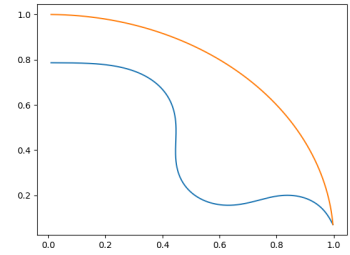
**Рис. 14:** Диаграмма  $z(0) - p$  для шарнирного закрепления.



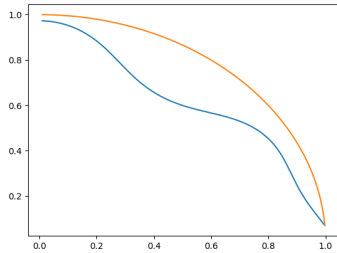
**Рис. 15:** Состояние А



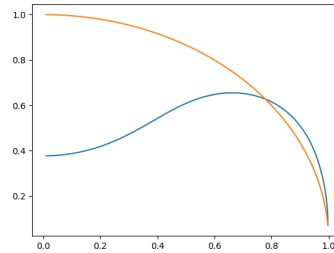
**Рис. 16:** Состояние В



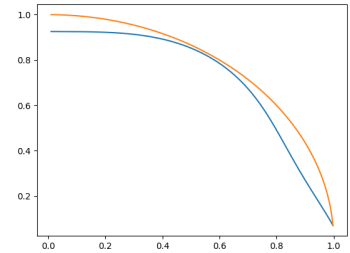
**Рис. 17:** Состояние С



**Рис. 18:** Состояние D



**Рис. 19:** Состояние E



**Рис. 20:** Состояние F

## Приложение Б

Импортируются библиотека `numpy`, функции `solve` и `inv` из пакета `scipy.linalg` для решения СЛАУ и обращения матрицы соответственно.

```
import numpy as np
from scipy.linalg import solve
from scipy.linalg import inv
```

Продолжение по параметру осуществляется с помощью увеличения на некоторый инкремент, в случае глобального параметра  $Pres$  при решении рассматриваемых задач принималось  $incr = -0.004$ . Значение  $\Delta$ , представляющее начало отрезка интегрирования должно выбираться из условия не превышения погрешности решения задачи Коши, в рассматриваемых задачах использовался метод Рунге-Кутты 4-го порядка, число точек ортогонализации принималось равным 100, поскольку отрезок интегрирования был равен  $L = 1.5$ , было взято  $h = 0.001$ .

Опустим ряд функций, отвечающих матрицам Якоби функций  $f(V, W, s)$  и  $g(V, W, s)$ , а также функции возвращающие значения недеформированной оболочки. Ортогонализацию Грамма-Шмидта производится с помощью функ-

ции  $\text{orth}(Y, N)$ , где матрице  $W$  соответствует матрица  $\Omega^{(s)}$ :

```
def orth(Y, N):
    z, W = np.zeros((K + 1, N)), np.zeros((K, K + 1))
    z[K] = Y[K]
    for i in range(K):
        z[i] = Y[i]
        for j in range(i):
            W[j, i] = Y[i] @ z[j]
            z[i] -= Y[i] @ z[j] * z[j]
    a = (z[i] @ Y[i]) ** 0.5
    W[i, i] = a
    z[i] /= a
    z[K] -= Y[K] @ z[i] * z[i]
    W[i, K] = Y[K] @ z[i]
return [z, W]
```

Функциями реализующими нахождение меридианальной кривизны  $\lambda_s$ , и всего вектора алгебраических переменных  $W$  не в граничной точке являются `res` и `res_f` соответственно:

```
def res(lam_s_0, lam_theta, rT_s, s):
    lam_n = lam_s_0
    lam_s_0 += 1
    c = mu*h_0(s)*ro_0(s)
    while abs(lam_n - lam_s_0)>1e-5:
        lam_s_0 = lam_n
        a = (lam_s_0 * lam_theta)**2
        f = a * lam_s_0 * (c * lam_s_0 - rT_s) - c
        f_der = a * (4 * c * lam_s_0 - 3 * rT_s)
        lam_n = lam_s_0 - f/f_der
    return lam_n
```

```

def res_f(lambda_s, lambda_theta, V, s):
    kappa_theta = np.sin(V[3]+phi_0(s))/
    (ro_0(s)*lambda_s*lambda_theta**2)-
    np.sin(phi_0(s))/ro_0(s)

    kappa_s = 3 * V[2] * lambda_s *
    (lambda_theta * lambda_s) ** 3 /
    ( ro_0(s) * mu * h_0(s) ** 3) - kappa_theta / 2

    T_theta = mu * h_0(s) *
    (lambda_theta ** 2 - 1/(lambda_s * lambda_theta) **
    lambda_theta

    M_theta = ((kappa_theta + kappa_s / 2) * mu * h_0(s)
    / (3 * lambda_theta * (lambda_theta * lambda_s)**3)

    return np.asarray([lambda_theta, lambda_s, T_theta,
    M_theta, kappa_s, kappa_theta])

```

Также вблизи нуля используется функция `sol_near_0`, определяющая в соответствие с изложенным правилом обработки граничного условия компоненты вектора  $V$ , при этом:

- $l_s$  - меридианальная кратность удлинений  
 - $\kappa$  - кривизна - $T_{th}$  - усилие

```

def sol_near_0(P_dif_0):
    a=P_dif_0[0]
    c=P_dif_0[2]

```



```
f=P_dif_0[5]
```

```
l_s = new_labda_near_s(h,a)
kappa = ((l_s**7) * c)/(mu*h_0(h)**3*ro_0(h))
d=h*(l_s**3*(kappa+1)-1)
e=h*(l_s*np.cos(d/2+h/2)-np.cos(h/2))
T_th = mu*h_0(h)*(l_s**6-1)/(l_s**5)
b=h*(T_th*np.sin(d/2+h/2) -
a*l_s**3*(kappa+1)/2-ro_0(h)*l_s**2*q_n(h))
P_dif=np.asarray([a,b,c,d,e,f])
```

```
return P_dif
```

Функцией реализующей решение уже линеаризованной задачи осуществляет функция `Bound_orth_par`, представляющая собой несколько усложненную версию ортогональной прогонки из-за алгебраических компонент. Данная функция вызывается на каждом новом шаге итерации, в качестве параметров ей передается:

А-матрица, соответствующая линеаризованной задаче  
 f-неоднородность + матрица  
 ff - исходная нелинейная задача  
 N-размерность  
 a,b-концы отрезка  
 Т-число, равноотстоящих узлов на сетке  
 Е-массив номеров свободных координат  
 Y\_n-решение на предыдущей итерации дифференциальной компоненты  
 W\_n-решение на предыдущей итерации алгебраической компоненты

```
def Bound_Orth_par(A, f, ff, N, a, b, T, E, Y_n, W_n):
```

```
K = len(E)
```

```
L = b - a
```

```
qq=L/T          ### Distance between points
```

```
Y0L, G, H, B = np.zeros((K + 1, N)), \
                np.zeros((T+1, K, K + 1)), \
                np.zeros((T+1, K + 1, N)), \
                np.zeros((T+1, K))
```

```
j = 0
for i in range(K):
    Y0L[i, E[j]] = 1 ###Linerly independent vectors
    j += 1
```

```
[Y0L, G[0]] = orth(Y0L, N)
H[0, :, :] = Y0L
```

```
for i in range(K):
    Y0L[i] = Euler_1(A, qq, a, Y0L[i], Y_n[0], W_n[0])
    Y0L[K] = Euler_1(f, qq, a, Y0L[K], Y_n[0], W_n[0])
```

```
[Y0L, G[1]] = orth(Y0L, N)
H[1, :, :] = Y0L
```

```
for x in range(1,T): ### From origin to the end
    p = RK(ff, qq, a + (x - 1) * qq, Y_n[x - 1], W_n[x - 1])
    q = RK(ff, qq / 2, a + x * qq, Y_n[x], W_n[x])
```

```

r = RK(ff , qq , a + x * qq , Y_n[x] , W_n[x])
for i in range(K):
    Y0L[i] = RK_adv(A, qq , a + x * qq , Y0L[i] ,
                    p ,
                    q ,
                    r ,W_n[x])
Y0L[K] = RK_adv(f , qq , a + x * qq , Y0L[K] ,
                p ,
                q ,
                r ,W_n[x])

[Y0L, G[x+1]] = orth(Y0L, N)
H[x+1, :, :] = Y0L

```

```

P_dif=np.zeros((T+1,N))
P_alg=np.zeros((T+1,N))

```

*##Boundary conditions*

```

B[T] = solve(H[T , :-1, K:].T, -H[T , -1, K:])

```

*### From end to origin*

```

for i in range(T, 0, -1):
    B[i-1] = solve(G[i, :, :-1], B[i] - G[i, :, -1])
    P_dif[i] = H[i, -1, :] + B[i] @ H[i, :-1, :]
    l_t = P_dif[i][4] / ro_0(b+(i-T)*qq) + 1
    l_s = res(W_n[i][1],l_t ,P_dif[i][0],b+(i-T)*qq)
    P_alg[i] = res_f(l_s ,l_t ,P_dif[i],b+(i-T)*qq)

```

```
P_dif[0]=np.array([B[0][0],B[0][1],B[0][2],0,0,0])
P_dif[0]=sol_near_0(P_dif[0])

return P_dif,P_alg
```